101791023

COPY

Docket No.: 03226/361001;
SUN040137
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Letters Patent of:
Andrew G. Tucker

Patent No.: 7,434,210

Issued: October 7, 2008

For: INTERPOSING LIBRARY FOR PAGE SIZE
DEPENDENCY CHECKING

**REQUEST FOR CERTIFICATE OF CORRECTION
PURSUANT TO 37 CFR 1.322**

Attention: Certificate of Correction Branch
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Upon reviewing the above-identified patent, Patentee noted typographical errors

which should be corrected.

In the Claims:

In Claim 3, column 8, line 26, "claim 1" should be --**claim 2**--.

In Claim 4, column 8, line 29, "claim 1" should be --**claim 2**--.

The errors were not in the application as filed by applicant; accordingly no fee is

required.

Transmitted herewith is a proposed Certificate of Correction effecting such

amendment.   Also enclosed, as evidence of the error, is a copy of the claims as issued, and a

436582.1

copy of the Claims as amended. Patentee respectfully solicits the granting of the requested

Certificate of Correction.

Applicant believes no fee is due with this request. However, if a fee is due,

please charge our Deposit Account No. 50-0591, under Order No. 03226/361001.

Dated: November 12, 2008   Respectfully submitted,

                                            # 63372

                                            By _____ ALY DOSA

                                            Robert P. Lord
                                            Registration No.: 46,479
                                            OSHA · LIANG LLP
                                            909 Fannin Street, Suite 3500
                                            Houston, Texas 77010
                                            (713) 228-8600
                                            (713) 228-8778 (Fax)

Embodiments of the invention provide a means to determine page size dependency of user-level applications. Specifically, by emulating a non-native page size using embodiments of the invention, a user can identify which user-level applications cease to function correctly when the non-native page size is emulated. In this manner, the user can efficiently determine which user-level applications are dependent on the native page size without any modifications to the native page size in the kernel.

The invention may be implemented on virtually any type of computer regardless of the platform being used. For example, as shown in FIG. 4, a networked computer system (40) includes a processor (42), associated memory (44), a storage device (46), and numerous other elements and functionalities typical of today's computers (not shown). The networked computer (40) may also include input means, such as a keyboard (48) and a mouse (50), and output means, such as a monitor (52). The networked computer system (40) is connected to a local area network (LAN) or a wide area network (e.g., the Internet) (not shown) via a network interface connection (not shown). Those skilled in the art will appreciate that these input and output means may take other forms. Further, those skilled in the art will appreciate that one or more elements of the aforementioned computer (40) may be located at a remote location and connected to the other elements over a network.

While the invention has been described with respect to a limited number of embodiments, those skilled in the art, having benefit of this disclosure, will appreciate that other embodiments can be devised which do not depart from the scope of the invention as disclosed herein. Accordingly, the scope of the invention should be limited only by the attached claims.

What is claimed is:

1. A method for processing a call comprising:

setting a non-native page size in an interposing library, wherein the interposing library is located in a user-level of a system and wherein the interposing library is located between a user-level application and a kernel, wherein the interposing library is generated by:

searching a plurality of interfaces to determine which of the plurality of interfaces include the native page size; and

modifying the plurality of interfaces that include the native page size to obtain a plurality of modified interfaces, wherein modifying the plurality of interfaces uses the non-native page size;

intercepting the call into the kernel by the interposing library, wherein the call is issued by the user-level application, wherein the call is dependent on the non-native page size and wherein the kernel uses a native page size;

modifying the call by the interposing library using a modified interface to obtain a modified call, wherein the modified call is dependent on the native page size, wherein the modified interface is one of the plurality of modified interfaces;

sending the modified call to the kernel;

generating a response to the modified call by the kernel using the native page size, wherein the response is dependent on the native page size;

sending the response to the user-level application;

intercepting the response by the interposing library;

modifying the response to obtain a modified response, wherein the modified response is dependent on the non-native page size; and

sending the modified response to the user-level application.

2. A system for checking page size dependency comprising:

a processor;

a kernel, located in a kernel-level of the system and executed on the processor, using a native page size;

a user-level application located in a user-level of the system; and

an interposing library located in the user-level configured to set a non-native page size to emulate and emulate the non-native page size to the user-level application, wherein the interposing library emulates the non-native page size by modifying results from the kernel based on the non-native page size, wherein the results from the kernel are based on the native page size, wherein the interposing library is further configured to modify a call dependent on the non-native page size from the user-level application to a call dependent on the native page size for the kernel, wherein the interposing library comprises a plurality of modified interfaces for emulating the non-native page size, wherein the plurality of modified interfaces are generated by searching a plurality of interfaces to determine which of the plurality of interfaces are dependent on the native page size and modifying the plurality of interfaces that include the native page size to obtain a plurality of modified interfaces.

3. The system of claim 1, wherein the interposing library uses a modified interface to emulate the non-native page size to the user-level application.

4. The system of claim 1, wherein the interposing library uses a modified interface to emulate the native page size to the kernel.

5. A computer system for checking page size dependency, comprising:

a processor;

a memory;

a storage device;

a computer display; and

software instructions stored in the memory for enabling the computer system under control of the processor, to:

set a non-native page size in an interposing library, wherein the interposing library is located in a user-level of a system and wherein the interposing library is located between a user-level application and a kernel, wherein the interposing library is generated by:

searching a plurality of interfaces to determine which of the plurality of interfaces are dependent on the native page size; and

modifying the plurality of interfaces that are dependent on the native page size to obtain a plurality of modified interfaces wherein modifying the plurality of interfaces uses the non-native page size;

intercept the call into the kernel by the interposing library, wherein the call is issued by the user-level application, wherein the call is dependent on the non-native page size and wherein the kernel uses a native page size;

modify the call by the interposing library using a modified interface to obtain a modified call, wherein the modified call is dependent on the native page size, wherein the modified interface is one of the plurality of modified interfaces;

send the modified call to the kernel;

generate a response to the modified call by the kernel using the native page size, wherein the response is dependent on the native page size;

send the response to the user-level application;

intercept the response by the interposing library;

NOV 1 9 2008.

modify the response to obtain a modified response, wherein the modified response is dependent on the non-native page size; and

send the modified response to the user-level application.

6. A network system having a plurality of nodes, comprising:

a processor;

a kernel, located in a kernel-level of the network system and executed on the processor, using a native page size;

a user-level application located in a user-level of the network system; and

an interposing library located in the user-level configured to set a non-native page size to emulate and emulate the non-native page size to the user-level application, wherein the interposing library emulates the non-native page size by modifying results from the kernel based on the non-native page size, wherein the results from the kernel are based on the native page size, wherein the

interposing library is further configured to modify a call dependent on the non-native page size from the user-level application to a call dependent on the native page size for the kernel, wherein the interposing library comprises a plurality of modified interfaces for emulating the non-native page size, wherein the plurality of modified interfaces are generated by searching a plurality of interfaces to determine which of the plurality of interfaces are dependent on the native page size and modifying the plurality of interfaces that include the native page size to obtain a plurality of modified interfaces,

wherein the kernel executes on any node of the plurality of nodes,

wherein the user-level application executes on any node of the plurality of nodes,

wherein the interposing library executes on any node of the plurality of nodes.

* * * * *

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A method for ~~checking page size dependency~~ processing a call comprising:

      setting a non-native page size in an interposing library, wherein the interposing library is located in a user-level of a system and wherein the interposing library is located between a user-level application and a kernel;

      ~~generating an interposing library comprising a first modified interface, wherein the first modified interface is dependent on a native page size;~~

      intercepting [[a]] the call into [[a]] the kernel by the interposing library, wherein the call is issued by the user-level application, wherein the call is dependent on [[a]] the non-native page size and wherein the kernel uses a native page size;

      modifying the call by the interposing library using ~~the first~~ a modified interface to obtain a modified call, wherein the modified call is dependent on the native page size;

      sending the modified call to the kernel; [[and]]

      generating a response to the modified call by the kernel using the native page size, wherein the response is dependent on the native page size;

      sending the response to the user-level application;

      intercepting the response by the interposing library;

      modifying the response to obtain a modified response, wherein the modified response is dependent on the non-native page size; and

      sending the modified response to the user-level application.

2. – 5 (Cancelled)

NOV 1 9 2008

6. (Currently Amended) The method of claim 1, <u>further comprising:</u>

~~wherein~~ generating the interposing library, <u>wherein generating the interposing library</u> comprises:

searching a plurality of interfaces to determine which of the plurality of interfaces include the native page size; and

modifying the plurality of interfaces that include the native page size to obtain a plurality of modified interfaces, wherein modifying the plurality of interfaces uses the non-native page size,

<u>wherein the modified interface is one of the plurality of modified interfaces.</u>

7. (Cancelled)

8. (Currently Amended) A system for checking page size dependency comprising:

a kernel<u>, located in a kernel-level of the system,</u> using a native page size;

a user-level application <u>located in a user-level of the system</u>; and

an interposing library<u> located in the user-level</u> configured to <u>set a non-native page size to emulate and</u> emulate [[a]] <u>the</u> non-native page size to the user-level application, wherein the interposing library emulates the non-native page size by modifying results from the kernel based on the non-native page size<u>, wherein the results from the kernel are based on the native page size.</u>

9. (Original) The system of claim 8, wherein the interposing library is further configured to modify a call dependent on the non-native page size from the user-level application to a call dependent on the native page size for the kernel.

10. (Original) The system of <u>claim 8,</u> wherein the interposing library uses a modified interface to emulate the non-native page size to the user-level application.

11. (Original) The system of <u>claim 8,</u> wherein the interposing library uses a modified interface to emulate the native page size to the kernel.

NOV 1 9 2008

12. (Original)   The system of claim 8, wherein the interposing library comprises a plurality of modified interfaces for emulating the non-native page size, wherein the plurality of modified interfaces are generated by searching a plurality of interfaces to determine which of the plurality of interfaces are dependent on the native page size and modifying the plurality of interfaces that include the native page size to obtain a plurality of modified interfaces.

13.   (Currently Amended) A computer system for checking page size dependency, comprising:

   a processor;

   a memory;

   a storage device;

   a computer display; and

   software instructions stored in the memory for enabling the computer system under control

      of the processor, to:

         set a non-native page size in an interposing library, wherein the interposing library is located in a user-level of a system and wherein the interposing library is located between a user-level application and a kernel;

         generate an interposing library comprising a first modified interface, wherein the first modified interface is dependent on a native page size;

         intercept [[a]] the call into [[a]] the kernel by the interposing library, wherein the call is issued by the user-level application, wherein the call is dependent on [[a]] the non-native page size and wherein the kernel uses a native page size;

         modify the call by the interposing library using the first a modified interface to obtain a modified call, wherein the modified call is dependent on the native page size;

         send the modified call to the kernel; [[and]]

         generate a response to the modified call by the kernel using the native page size, wherein the response is dependent on the native page size;

         send the response to the user-level application;

         intercept the response by the interposing library;

NOV 1 9 2008.

modify the response to obtain a modified response, wherein the modified response is

dependent on the non-native page size; and

send the modified response to the user-level application.

14. -17 (Cancelled)

18. (Currently Amended) The computer system of claim 13, wherein software instructions further

comprise:

software instructions for generating the interposing library, wherein the software instructions

comprise:

software instructions to search a plurality of interfaces to determine which of the

plurality of interfaces are dependent on the native page size; and

software instructions to modify the plurality of interfaces that are dependent on the

native page size to obtain a plurality of modified interfaces, wherein the

modified interface is one of the plurality of modified interfaces.

19. (Cancelled)

20. (Currently Amended) A network system having a plurality of nodes, comprising:

a kernel, located in a kernel-level of the system, using a native page size;

a user-level application located in a user-level of the system; and

an interposing library located in the user-level configured to set a non-native page size to

emulate and emulate [[a]] the non-native page size to the user-level application,

wherein the interposing library emulates the non-native page size by modifying

results from the kernel based on the non-native page size, wherein the results from

the kernel are based on the native page size,

wherein the kernel executes on any node of the plurality of nodes,

wherein the user-level application executes on any node of the plurality of nodes,

wherein the interposing library executes on any node of the plurality of nodes.

NOV 1 9 2008

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

Page _1_ of _1_

PATENT NO.   :   7,434,210

APPLICATION NO. :   10/791,023

ISSUE DATE   :   October 7, 2008

INVENTOR(S)   :   Andrew G. Tucker

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In the Claims:

In Claim 3, column 8, line 26, "claim 1" should be --claim 2--.

In Claim 4, column 8, line 29, "claim 1" should be --claim 2--.

MAILING ADDRESS OF SENDER (Please do not use customer number below):
Aly Z. Dossa
OSHA · LIANG LLP   1
909 Fannin Street, Suite 3500
Houston, Texas 77010

NOV 1 9 2008

436584.1

Application No. (if known): 10/791,023   Attorney Docket No.: 03226/361001; SUN040137

Certificate of Mailing under 37 CFR 1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to:

> Attention: Certificate of Correction Branch
> Commissioner for Patents
> P.O. Box 1450
> Alexandria, VA 22313-1450

on _____November 12, 2008_____ .
　　　　　　　　Date

_____
　　　　　　　　　　　　　　　　　Signature

Blanca E. Ramos
_____
Typed or printed name of person signing Certificate

_____   ____(713) 228-8600____
Registration Number, if applicable   Telephone Number

Note: Each paper must have its own certificate of mailing, or this certificate must identify each submitted paper.

Request for Certificate of Correction (No Fee) with attachments (8 pages)
Certificate of Correction (1 page)
Return Receipt Post Card

NOV 1 9 2008.